ISSN 1000-9825, CODEN RUXUEW
Journal of Software, Vol.21, No.1, January 2010, pp.1−13
doi: 10.3724/SP.J.1001.2010.03494
© by Institute of Software, the Chinese Academy of Sciences. All rights reserved.
E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

一类扩展的动态描述逻辑*

常　亮[1,2,3]，史忠植[1+]，陈立民[1,3]，牛温佳[1,3]

[1](中国科学院 计算技术研究所 智能信息处理重点实验室,北京 100190)
[2](桂林电子科技大学 计算机与控制学院,广西 桂林 541004)
[3](中国科学院 研究生院,北京 100049)

Family of Extended Dynamic Description Logics

CHANG Liang[1,2,3], SHI Zhong-Zhi[1+], CHEN Li-Min[1,3], NIU Wen-Jia[1,3]

[1](Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, The Chinese Academy of Sciences, Beijing 100190, China)
[2](School of Computer and Control, Guilin University of Electronic Technology, Guilin 541004, China)
[3](Graduate University, The Chinese Academy of Sciences, Beijing 100049, China)
+ Corresponding author: E-mail: shizz@ics.ict.ac.cn

Chang L, Shi ZZ, Chen LM, Niu WJ. Family of extended dynamic description logics. *Journal of Software*, 2010,21(1):1−13. http://www.jos.org.cn/1000-9825/3494.htm

Abstract: As the extension of the description logic, many dynamic description logics are proposed for modeling and reasoning about Semantic Web Services. These dynamic description logics provide decidable reasoning mechanisms for the result of action execution. However, the procedure of action execution can not be described and reasoned about according to these logics. Inspired by an extended propositional dynamic logic studied by Pratt, this paper proposes two improvements to the dynamic description logic. One is that, semantics of actions in the dynamic description logic are redefined in such a way that each action is interpreted as a set of trajectories, where each trajectory is a sequence of possible worlds of the semantic model. The other is that, assertions on the procedure of action execution are introduced to the logic so that not only the result but also the procedure can be described for the execution of actions. As a result, a family of extended dynamic description logics named *EDDL*($\mathcal{X}$) is presented in this paper, where $\mathcal{X}$ represents well-studied description logics ranging from ALC (attributive language with complements) to *SHOIN*(***D***). Taking the description logic ALCQO (attributive language with complements, qualified number restrictions and nominals) as an example of the $\mathcal{X}$ of *EDDL*($\mathcal{X}$), this paper proposes a tableau decision algorithm for the logic EDDL(ALCQO) and proves that this algorithm is terminating, sound and complete. With *EDDL*($\mathcal{X}$), both the result and the procedure of action execution can be described and reasoned about. Therefore, compared with dynamic description logics, *EDDL*($\mathcal{X}$) provides further support for modeling and reasoning about semantic Web services.
Key words: dynamic description logic; reasoning about actions; procedure of action execution; tableau decision algorithm; semantic Web service

* Supported by the National Natural Science Foundation of China under Grant Nos.90604017, 60775035, 60803033 (国家自然科学基金); the National High-Tech Research and Development Plan of China under Grant No.2007AA01Z132 (国家高技术研究发展计划(863)); the National Basic Research Program of China under Grant No.2007CB311004 (国家重点基础研究发展计划(973))
Received 2008-01-22; Revised 2008-07-14; Accepted 2008-10-07; Published online 2009-06-05

© 中国科学院软件研究所 http://www.c-s-a.org.cn

摘　要: 作为描述逻辑的扩展,动态描述逻辑为语义 Web 服务的建模和推理提供了一种有效途径.在将语义 Web 服务建模为动作之后,动态描述逻辑从动作执行结果的角度提供了丰富的推理机制,但对于动作的执行过程却不能加以处理.借鉴 Pratt 关于命题动态逻辑的相关研究,一方面,对动态描述逻辑中动作的语义重新进行定义,将每个动作解释为由关于可能世界的序列组成的集合;另一方面,在动态描述逻辑中引入动作过程断言,用来对动作的执行过程加以刻画.在此基础上提出一类扩展的动态描述逻辑 *EDDL*(*X*),其中的 *X* 表示从 ALC(attributive language with complements)到 *SHOIN*(***D***)等具有不同描述能力的描述逻辑.以 *X* 为描述逻辑 ALCQO(attributive language with complements,qualified number restrictions and nominals)的情况为例,给出了 EDDL(ALCQO)的表判定算法,并证明了算法的可终止性、可靠性和完备性.*EDDL*(*X*)可以从动作执行过程和动作执行结果两个方面对动作进行全面的刻画和推理,为语义 Web 服务的建模和推理提供了进一步的逻辑支持.

关键词: 动态描述逻辑;动作推理;动作的执行过程;表判定算法;语义 Web 服务
中图法分类号: TP301　　**文献标识码**: A

语义Web是万维网发展的一个重要趋势,旨在实现Web上数据之间的链接,并且为这些数据赋予语义信息,使得计算机能够理解和自动处理.描述逻辑在语义Web中扮演着关键角色,是W3C推荐的Web本体语言OWL(Web ontology language)的逻辑基础[1].描述逻辑的主要特点在于既具有较强的描述能力又具有可判定性,并且存在有效的判定算法和推理机制作为支撑.针对Web环境的各种特征和应用需求,研究者提出了描述逻辑的各种扩展,以期为语义Web提供更为充分的逻辑支持.例如,针对Web环境的开放性提出了非单调描述逻辑[2]和多值描述逻辑[3],针对Web应用中需要处理的模糊信息提出了模糊描述逻辑[4−6]等.

Web服务是一类典型的Web应用,其与语义Web的结合产生了语义Web服务[7].语义Web服务克服了Web服务在语义操作能力上的局限,有望解决Web服务的自动发现和自动组合问题.在关于语义Web服务的研究中,一个基本问题是如何将语义Web上基于描述逻辑的静态的领域知识与关于Web服务动态功能的描述有机地结合起来.

针对描述逻辑只能处理静态领域知识的这一局限,文献[8]将描述逻辑ALC(attributive language with complements)、动态逻辑以及动作理论有机地结合起来,提出了一种动态描述逻辑DDL(dynamic description logic).DDL可以在一个逻辑系统内对基于描述逻辑的静态的领域知识以及关于动作的知识进行统一的描述和推理.文献[9]在DDL的基础上采用可能模型途径来定义原子动作的语义,针对描述逻辑ALCO(attributive language with complements and nominals),ALCQO(attributive language with complements,qualified number restrictions and nominals)和ALCQIO(attributive language with complements,qualified number restrictions,inverse roles and nominals)分别构建出D-ALCO,D-ALCQO以及D-ALCQIO等具有不同描述能力的动态描述逻辑,并为这些动态描述逻辑依次提供了适用于开世界假设的表判定算法.文献[10]基于动态描述逻辑研究了关于动作的各种推理问题,将动作的可实现性、可执行性、投影、规划等问题转换为动态描述逻辑中公式的可满足性问题,从而可以基于判定算法在信息不完全的情况下进行推理.动态描述逻辑为语义Web服务的建模和推理以及在此基础上的服务发现和服务组合提供了一种有效的途径和工具[11].

动态描述逻辑将每个动作解释为关于可能世界的二元关系,体现了可能世界之间在动作的作用下是否可达;这种语义解释限制了动态描述逻辑的描述和推理能力,不能对动作的执行过程进行刻画和推理.针对这一局限,借鉴文献[12]关于命题动态逻辑的相关工作,本文在文献[9]的基础上对动态描述逻辑中动作的语义重新进行定义,将每个动作解释为由关于可能世界的序列组成的集合,进而引入形如$\{\pi\}\varphi$的动作过程断言;同时,本文对这种构建动态描述逻辑的方式进一步推广,得到一类扩展的动态描述逻辑 *EDDL*(*X*),其中的 *X* 表示从 ALC 到 *SHOIN*(***D***)等具有不同描述能力的描述逻辑.以 *X* 为描述逻辑 ALCQO 的情况为例,本文给出 EDDL(ALCQO)的表判定算法,并证明了算法的正确性.最后,本文以一个销售图书和 CD 的 Web 服务系统为例,应用 EDDL(ALCQO)对其中的 Web 服务进行建模,然后从 Web 服务执行结果以及 Web 服务执行过程两个方面进行推理.

1 扩展的动态描述逻辑 *EDDL*($\mathcal{X}$)

本节给出 *EDDL*($\mathcal{X}$)的语法和语义定义,其中的 $\mathcal{X}$ 代表具有不同描述能力的描述逻辑,可以从最典型的描述逻辑 ALC 到与本体语言 OWL DL 对应的描述逻辑 *SHOIN*(***D***)等.

EDDL($\mathcal{X}$)的基本符号包括:(1) 描述逻辑$\mathcal{X}$中的基本符号,又包括 4 个部分:由概念名组成的集合N_C,由角色名组成的集合N_R,由个体名组成的集合N_I以及由描述逻辑$\mathcal{X}$中的概念构造符和角色构造符组成的集合$N_S(\mathcal{X})$.例如,当$\mathcal{X}$为描述逻辑ALCQO时,有$N_S(ALCQO)=\{\neg,\sqcup,\forall,\leq\}$;(2) 由原子动作名组成的集合$N_A$;(3) 概念构造符{}和@;(4) 公式构造符[],{},¬和∨;(5) 动作构造符∪,;,*和?;(6) 其他符号,包括定义号“≡”、圆括号“()”及逗号“,”.

从这些符号出发可以递归地构造出角色、概念、公式以及动作.

定义 1(角色). R 是 *EDDL*($\mathcal{X}$)中的角色当且仅当 R 是 $\mathcal{X}$ 所代表的描述逻辑中的角色.

例如,当$\mathcal{X}$为描述逻辑ALCQO时,有如下结论:R是EDDL(ALCQO)中的角色当且仅当$R\in N_R$.

定义 2(概念). 对 *EDDL*($\mathcal{X}$)中的概念递归定义如下:

(1) 通过描述逻辑 $\mathcal{X}$ 中的语法规则构造而得的任一概念都是 *EDDL*($\mathcal{X}$)中的概念;

(2) 如果 u 为个体名,则$\{u\}$是 *EDDL*($\mathcal{X}$)中的概念;

(3) 如果u为个体名,C为概念,则$@_uC$也是*EDDL*($\mathcal{X}$)中的概念.

例如,当 $\mathcal{X}$ 为描述逻辑 ALCQO 时,EDDL(ALCQO)中的概念由如下产生式生成:

$$C,D::=C_i|\neg C|C\sqcup D|\forall R.C|\leq nR.C|\{u\}|@_uC,$$

其中,$C_i\in N_C,u\in N_I,R$为角色,n为非负整数.

此外,也可以引入形如 $C\sqcap D,\exists R.C,\geq nR.C,\top$以及$\bot$的概念,分别作为$\neg(\neg C\sqcup\neg D),\neg(\forall R.\neg C),\neg(\leq(n-1)R.C)$, $C\sqcup\neg C$ 和 $C\sqcap\neg C$ 的缩写.

令C_i为概念名,D为概念,则称$C_i\equiv D$为概念定义式.对于由概念定义式组成的有限集合$\mathcal{T}$,如果每个概念名最多在$\mathcal{T}$中某个概念定义式的左边出现 1 次,则称$\mathcal{T}$为TBox.

相对于某个TBox $\mathcal{T}$,如果某个概念名C_i出现在$\mathcal{T}$中概念定义式的左边,则称C_i为被定义的概念名,否则称其为简单概念名.

定义 3(公式). *EDDL*($\mathcal{X}$)中的公式由如下产生式生成:

$$\varphi,\psi::=C(u)|R(u,v)|[\pi]\varphi|\{\pi\}\varphi|\neg\varphi|\varphi\vee\psi,$$

其中,$u,v\in N_I,C$为概念,R为角色,π为动作.将形如$[\pi]\varphi$和$\{\pi\}\varphi$的公式分别称为动作结果断言和动作过程断言.

此外,也可以引入形如$\langle\pi\rangle\varphi,\varphi\wedge\psi,\varphi\rightarrow\psi$,true 以及 false 的公式,分别作为$\neg[\pi]\neg\varphi,\neg(\neg\varphi\vee\neg\psi),\neg\varphi\vee\psi,\varphi\vee\neg\varphi$以及$\varphi\wedge\neg\varphi$的缩写.

给定某个TBox $\mathcal{T}$,如果C为简单概念名,$u,v\in N_I,R$为角色,则将形如$C(u),\neg C(u),R(u,v)$以及$\neg R(u,v)$的公式都称为简单公式.

定义 4(动作). *EDDL*($\mathcal{X}$)中的动作由如下产生式生成:

$$\pi,\pi'::=\alpha|\varphi?|\pi\cup\pi'|\pi;\pi'|\pi^*,$$

其中,$\alpha\in N_A,\varphi$为公式.将形如$\alpha,\varphi?,\pi\cup\pi',\pi;\pi'$以及$\pi^*$的动作分别称为原子、测试、选择、顺序以及迭代动作.

基于上述动作,可以进一步刻画出复杂的控制结构.例如,可以将条件选择结构“if φ then π else π'”刻画为“$(\varphi?;\pi)\cup((\neg\varphi)?;\pi')$”,将循环控制结构“while φ do π”刻画为“$(\varphi?;\pi)^*;(\neg\varphi)?$”,将循环控制结构“do π until φ”刻画为“$\pi;((\neg\varphi)?;\pi)^*;\varphi?$”等.

定义 5(原子动作定义式). 相对于某个TBox $\mathcal{T}$,将$\alpha\equiv(P,E)$称为一个原子动作定义式,其中:(1) $\alpha\in N_A$,是所定义的原子动作;(2) P是由公式组成的有限集合,表示执行动作之前必须满足的前提条件;(3) E是由简单公式组成的有限集合,表示执行动作后将会产生的影响.

对于由原子动作定义式组成的有限集合 $\mathcal{A}c$,如果每个原子动作名最多在 $\mathcal{A}c$ 中某个定义式的左边出现 1 次,则称 $\mathcal{A}c$ 为原子动作 Box,简称为 ActBox.

相对于某个 TBox $\mathcal{T}$和 ActBox $\mathcal{A}c$,如果原子动作α出现在$\mathcal{A}c$中某个定义式的左边,则称α为被定义的原子动

© 中国科学院软件研究所 http://www.c-s-a.org.cn

作.并且,如果关于α的原子动作定义式形如$\alpha\equiv(P,E)$,则也可以用二元组(P,E)指代原子动作α.

在语义方面,$EDDL(X)$在描述逻辑 X 的基础上引入了动态维,使得语义模型从整体上体现为由多个可能世界构成的空间.在其中的每个可能世界下分别对概念名、角色名以及个体名进行解释;同时,每个动作被解释为由关于可能世界的序列组成的集合,体现了动作执行过程中经历的关于可能世界的所有轨迹.

定义 6(模型). $EDDL(X)$模型为一个三元组$M=(\Sigma,\Delta_M,I)$,其中:

(1) Σ是形如$\Sigma=(W,\cdot^T)$的框架,在该框架中:(i) W是由可能世界组成的非空集合,并且含有一个特殊的可能世界Λ;(ii) 函数$\cdot^T$将N_A中的每个原子动作名α_i映射为W上的某个二元关系$\alpha_i^T\subseteq W\times W$,并且满足:对于$W$中任一可能世界$w$,都有$(w,\Lambda)\in\alpha_i^T$;

(2) Δ_M是由个体组成的非空集合,作为该模型的论域;

(3) 函数I对W中除Λ之外的每个可能世界w赋予一个解释$I(w)=(\Delta_M,\cdot^{I(w)})$,其中的解释函数$\cdot^{I(w)}$满足以下条件:(i) 将$N_C$中的每个概念名$C_i$解释为$\Delta_M$的某个子集$C_i^{I(w)}\subseteq\Delta_M$;(ii) 将$N_R$中的每个角色名$R_i$解释为$\Delta_M$上的某个二元关系$R_i^{I(w)}\subseteq\Delta_M\times\Delta_M$;(iii) 将$N_I$中的每个个体名$p_i$解释为$\Delta_M$中的某个元素$p_i^{I(w)}\in\Delta_M$,并且满足:对于$W$中任一可能世界$w'$都有$p_i^{I(w)}=p_i^{I(w')}$;由于$p_i$的解释与可能世界无关,因而也将$p_i^{I(w)}$简记为$p_i^I$.

与动态描述逻辑的语义模型相比,上述定义在每个模型 M 中都引入了一个特殊的可能世界Λ,其目的是保证对任一动作的解释都是相对于 W 为完全的.下文将会对此作进一步的阐述.

定义 7(轨迹). 相对于某个模型$M=(\Sigma,\Delta_M,I)$(其中$\Sigma=(W,\cdot^T)$),称由W中的可能世界构成的任一序列$(w_1,w_2,w_3,\ldots)$为关于M的一个轨迹.

对于任一轨迹τ,分别用$|\tau|$和$\tau[i]$表示τ的长度以及τ中的第 i 个可能世界.因此,可以用$\tau[1]$表示τ中的第 1 个可能世界;如果τ的长度有限,则可以用$\tau[|\tau|]$表示τ中的最后一个可能世界.

定义 8(轨迹的混合积). 令T_1,T_2是由轨迹组成的两个集合;对它们的混合积$T_1\times T_2$定义如下:

$T_1\times T_2:=\{(\tau_1[1],\tau_1[2],\ldots,\tau_1[|\tau_1|],\tau_2[2],\ldots,\tau_2[|\tau_2|])|\tau_1\in T_1,\tau_2\in T_2,\tau_1$的长度有限并且$\tau_1[|\tau_1|]=\tau_2[1]\}$.

该定义中,对分别来自T_1和T_2的任意一对轨迹τ_1和τ_2,若τ_1的长度有限且τ_1的最后一个可能世界与τ_2的第 1 个可能世界相同,则将这两个轨迹串联起来,去掉其中重复的可能世界$\tau_2[1]$,然后将其作为$T_1\times T_2$的一个元素.

定义 9(轨迹集合的完全性). 令T是由关于模型$M=(\Sigma,\Delta_M,I)$(其中$\Sigma=(W,\cdot^T)$)的轨迹组成的任一集合;如果对于W中的任一可能世界w来说都存在某个轨迹$\tau\in T$使得$\tau[1]=w$,则称T相对于W是完全的.

在上述定义和符号的基础上,可以给出 $EDDL(X)$的语义定义.

定义 10(语义). 令$M=(\Sigma,\Delta_M,I)$为$EDDL(X)$模型,其中$\Sigma=(W,\cdot^T)$;对$EDDL(X)$中概念、公式和动作的语义归纳定义如下:

首先,相对于W中除了Λ之外的任一可能世界w,将任一角色R解释为Δ_M上的某个二元关系$R^{I(w)}$,将任一概念C解释为Δ_M的某个子集$C^{I(w)}$.归纳定义如下:

(1) $\{u\}^{I(w)}:=\{u^I\}$,其中,$u\in N_I$;

(2) 如果$u^{I(w)}\in C^{I(w)}$,则$(@_uC)^{I(w)}:=\Delta_M$,否则$(@_uC)^{I(w)}:=\varnothing$,其中的$\varnothing$为空集;

对于其他形式的概念或角色A,只需要将描述逻辑X中形如A^I的解释扩展为$A^{I(w)}$的形式,从而将该解释与可能世界w关联起来.例如,如果X为描述逻辑ALCQO,则相应地增加以下语义定义:

(3) $(\neg C)^{I(w)}:=\Delta_M\backslash C^{I(w)}$,其中的\为集合差运算;

(4) $(C\sqcup D)^{I(w)}:=C^{I(w)}\cup D^{I(w)}$,其中的$\cup$为集合并运算;

(5) $(\forall R.C)^{I(w)}:=\{x|$对于任一$y\in\Delta_M$:如果$(x,y)\in R^{I(w)}$,则必然有$y\in C^{I(w)}\}$;

(6) $(\le nR.C)^{I(w)}:=\{x|\#\{y|(x,y)\in R^{I(w)}$并且$y\in C^{I(w)}\}\le n\}$,其中的#表示集合的元素个数.

其次,对于 W 中除了Λ之外的任一可能世界 w,用$(M,w)\vDash\varphi$表示公式φ在模型 M 中的可能世界 w 下成立.根据φ的结构归纳定义如下:

(7) $(M,w)\vDash C(u)$ iff $u^I\in C^{I(w)}$;

© 中国科学院软件研究所 http://www.c-s-a.org.cn

(8) $(M,w)\vDash R(u,v)$ iff $(u^I,v^I)\in R^{I(w)}$;

(9) $(M,w)\vDash\neg\varphi$ iff $(M,w)\nvDash\varphi$(即公式φ在模型 M 中的可能世界 w 下不成立);

(10) $(M,w)\vDash\varphi\vee\psi$ iff $(M,w)\vDash\varphi$或者$(M,w)\vDash\psi$;

(11) $(M,w)\vDash[\pi]\varphi$ iff 对于任一轨迹$\tau\in\pi^T$:如果$\tau[1]=w$并且$\tau[|\tau|]\neq\Lambda$,则必然有$(M,\tau[|\tau|])\vDash\varphi$;

(12) $(M,w)\vDash\{\pi\}\varphi$ iff 对于任一轨迹$\tau\in\pi^T$:如果$\tau[1]=w$,则对于任一 $1\le i\le|\tau|$来说,在$\tau[i]\neq\Lambda$时都有$(M,\tau[i])\vDash\varphi$.

最后,对于任一动作π,将其映射为由关于M的轨迹组成的某个集合π^T.根据π的结构归纳定义如下:

(13) $(\varphi?)^T:=\{(w,w)|(M,w)\vDash\varphi\}\cup\{(w,\Lambda)|(M,w)\nvDash\varphi\}\cup\{(\Lambda,\Lambda)\}$;

(14) $(\pi\cup\pi')^T:=\pi^T\cup\pi'^T$;

(15) $(\pi;\pi')^T:=\pi^T\times\pi'^T$,其中的$\times$为关于轨迹集合的混合积运算;

(16) $(\pi^*)^T:=(\pi^0)^T\cup(\pi^1)^T\cup(\pi^2)^T\cup\ldots$,其中,$(\pi^0)^T:=\{(w,w)|w\in W\}$,对于任一$i\ge1$ 都有$(\pi^i)^T:=(\pi^{i-1};\pi)^T$.

显然,根据上述定义,对任一测试动作φ?的解释都是相对于 W 为完全的;同时,根据定义 6,对任一原子动作的解释也是相对于 W 为完全的.再根据上述定义中的(14)~(16),保证了对任一动作的解释都是相对于 W 为完全的.这种性质将在很大程度上简化本文第 3 节对引理 2 的证明.

定义 11(TBox的模型). 令$M=(\Sigma,\Delta_M,I)$为$EDDL(\mathcal{X})$模型,其中$\Sigma=(W,\cdot^T)$,则,

(1) M是某个概念定义式$C_i\equiv D$的模型,记为$M\vDash C_i\equiv D$,当且仅当对于任一$w\in W$都有 $C_i^{I(w)}=D^{I(w)}$;

(2) M是某个TBox $\mathcal{T}$的模型,记为$M\vDash\mathcal{T}$,当且仅当对于任一$C_i\equiv D\in\mathcal{T}$都有$M\vDash C_i\equiv D$.

定义 12(ActBox的模型). 令$M=(\Sigma,\Delta_M,I)$为$EDDL(\mathcal{X})$模型,其中$\Sigma=(W,\cdot^T)$,则,

(1) M是某个原子动作定义式$\alpha\equiv(P,E)$的模型,记为$M\vDash\alpha\equiv(P,E)$,当且仅当$\alpha^T=\{(w,\Lambda)|w\in W\}\cup\{(w,w')|$:① 对于任一$\psi\in P$都有$(M,w)\vDash\psi$;② 对于任一$\varphi\in E$都有$(M,w)\vDash\neg\varphi$;③ 对于任一简单概念名$C$都有$C^{I(w')}=(C^{I(w)}\cup\{u^I|C(u)\in E\})\backslash\{u^I|\neg C(u)\in E\}$,以及对于任一角色名$R$都有$R^{I(w')}=(R^{I(w)}\cup\{(u^I,v^I)|R(u,v)\in E\})\backslash\{(u^I,v^I)|\neg R(u,v)\in E\}\}$;

(2) M 是某个 ActBox $\mathcal{Ac}$ 的模型,记为 $M\vDash\mathcal{Ac}$,当且仅当对于任一$\alpha\equiv(P,E)\in\mathcal{Ac}$ 都有 $M\vDash\alpha\equiv(P,E)$.

上面关于原子动作定义式$\alpha\equiv(P,E)$的语义定义采用了可能模型途径(possible models approach)[13].对于等式右边的第 2 个集合来说,其中的条件①保证了P中的公式在执行原子动作之前必须得以满足;条件②和条件③则保证了E是执行原子动作之后产生的影响,并且这种影响体现为概念名和角色名的解释所发生的变化,进而体现为可能世界之间的转换关系.

定义 13(公式的可满足性). 令$\mathcal{T},\mathcal{Ac}$分别为TBox和ActBox,φ为公式,并且φ中出现的各个原子动作都是相对于$\mathcal{T}$和$\mathcal{Ac}$被定义的.称公式φ相对于$\mathcal{T}$和$\mathcal{Ac}$是可满足的,当且仅当存在某个模型$M=(\Sigma,\Delta_M,I)$以及该模型中的某个可能世界w,使得$M\vDash\mathcal{T},M\vDash\mathcal{Ac}$和$(M,w)\vDash\varphi$.

公式的可满足性问题是 $EDDL(\mathcal{X})$中最基本的推理问题.下面以 $\mathcal{X}$ 为描述逻辑 ALCQO 的情况为例,给出EDDL(ALCQO)的表判定算法.

2 EDDL(ALCQO)的表判定算法

令 $\mathcal{T},\mathcal{Ac}$ 分别为 TBox 和 ActBox.下面先引入一系列符号和术语.

首先,对于任一公式φ,用$nf_{\mathcal{T},\mathcal{Ac}}(\varphi)$表示经过如下转换后得到的结果:① 对于$\varphi$中出现的每个被定义的概念名,用$\mathcal{T}$中相应概念定义式的右边的概念进行替换;② 对于$\varphi$中出现的每个原子动作,根据$\mathcal{Ac}$中相应的原子动作定义式将其改写为由前提条件$P$和所产生的影响$E$组成的二元组形式;③ 反复进行上述替换过程,直到不出现被定义的概念名,并且每个原子动作都已经改写为二元组形式.与文献[9]相同,本文假设TBox $\mathcal{T}$与ActBox $\mathcal{Ac}$中不存在循环定义,因而上述转换过程总是可终止的.

其次,对于任一简单公式φ,如果φ形如$\neg R(u,v)$或$\neg C(u)$,则用$\varphi^{\neg}$表示相应的简单公式$R(u,v)$和$C(u)$;如果φ形如$R(u,v)$或$C(u)$,并且C为简单概念名,则用$\varphi^{\neg}$相应地表示$\neg R(u,v)$和$\neg C(u)$.在此基础上,对于由简单公式组成的任一集合E,用$E^{\neg}$表示集合$\{\varphi^{\neg}|\varphi\in E\}$.

接下来我们引入前缀和带前缀的公式这两个概念.

定义 14(前缀). 任一前缀$\sigma.\varepsilon$是一个由顺序动作σ以及关于简单公式的集合ε组成的二元组,并且满足如下产生式规则:

$$\sigma.\varepsilon::=(\varnothing,\varnothing).\varnothing|\sigma;(P,E).(\varepsilon\backslash E^{\neg})\cup E,$$

其中,(P,E)为原子动作,$\sigma;(P,E)$是新生成的顺序动作,$(\varepsilon\backslash E^{\neg})\cup E$是新生成的由简单公式组成的集合.也将$(\varnothing,\varnothing).\varnothing$称为初始前缀,表示为$\sigma_0.\varepsilon_0$.

令$\sigma.\varepsilon$为前缀,φ为公式,则将$\sigma.\varepsilon:\varphi$称为带前缀的公式.

最后引入分枝、分枝的扩展、饱和的分枝、可忽略的分枝以及存在冲突的分枝这几个概念.

定义 15(分枝). 任一分枝$\mathcal{B}$是由以下 4 类元素组成的有限集合:(1) 带前缀的公式;(2) 形如$X\equiv\neg[\pi^*]\varphi$的可能性标记,其中的$[\pi^*]\varphi$是由迭代动作$\pi^*$和公式$\varphi$构成的动作结果断言,$X$为字符串;(3) 形如$u=v$的关于个体名的等式,其中,$u,v\in N_I$;(4) 形如$u\neq v$的关于个体名的不等式,其中,$u,v\in N_I$.

对于任一分枝$\mathcal{B}$,用$\mathcal{B}_{Ind}$表示由$\mathcal{B}$中关于个体名的所有等式及不等式组成的集合.在此基础上,与文献[9]相同,用$\mathcal{B}_{Ind}^*$表示$\mathcal{B}_{Ind}$关于等价关系“=”和对称关系“≠”的闭包.此外,对于任一个体名u,用$[u]_{=\mathcal{B}}$表示u在$\mathcal{B}$中关于等价关系“=”的等价类,即$[u]_{=\mathcal{B}}:=\{v|u=v\in\mathcal{B}_{Ind}^*\}$.

对于任一分枝$\mathcal{B}$,如果它满足表 1~表 5 中某条扩展规则的前件,则可以应用该扩展规则对$\mathcal{B}$进行扩展,在$\mathcal{B}$中加入带前缀的公式、可能性标记以及关于个体名的等式或不等式.

定义 16(饱和的分枝). 如果表 1~表 5 中所有扩展规则都不能对$\mathcal{B}$进行扩展,则称$\mathcal{B}$为饱和的.

定义 17(可忽略的分枝). 如果分枝$\mathcal{B}$是饱和的并且含有未被实现的可能性标记,则称$\mathcal{B}$是可忽略的,否则称$\mathcal{B}$为不可忽略的.其中,称某个可能性标记$X\equiv\neg[\pi^*]\varphi$在$\mathcal{B}$中是被实现的当且仅当存在某两个前缀$\sigma.\varepsilon$和$\sigma'.\varepsilon'$,使得$\varepsilon=\varepsilon'$,$\sigma.\varepsilon:X\in\mathcal{B}$以及$\sigma'.\varepsilon':\neg\varphi\in\mathcal{B}$.

定义 18(存在冲突的分枝). 如果分枝$\mathcal{B}$中至少出现以下一种情况,则称$\mathcal{B}$存在冲突,否则称$\mathcal{B}$为无冲突的:

(1) 存在某个公式φ以及某两个前缀$\sigma.\varepsilon$和$\sigma'.\varepsilon'$,使得$\varepsilon=\varepsilon'$,$\sigma.\varepsilon:\varphi\in\mathcal{B}$以及$\sigma'.\varepsilon':\neg\varphi\in\mathcal{B}$;

(2) 存在某个概念C、某个个体名u以及某两个前缀$\sigma.\varepsilon$和$\sigma'.\varepsilon'$,使得$\varepsilon=\varepsilon'$,$\sigma.\varepsilon:C(u)\in\mathcal{B}$以及$\sigma'.\varepsilon':(\neg C)(u)\in\mathcal{B}$;

(3) 存在某个个体名u,使得$u\neq u\in\mathcal{B}_{Ind}^*$.

表 1~表 5 的扩展规则直观地体现了 EDDL(ALCQO)的语义定义,是表判定算法的核心部分.其中,表 1 和表 2 中的扩展规则与动作无关,因而与文献[9]中关于 D-ALCQO 的概念的扩展规则类似.

Table 1 Tableau expansion rules on EDDL(ALCQO) concepts

表 1 EDDL(ALCQO)中关于概念的扩展规则

Name	Rule
$\neg\neg_C$-rule	If $\sigma.\varepsilon:(\neg(\neg C))(x)\in\mathcal{B}$, and $\sigma.\varepsilon:C(x)\notin\mathcal{B}$, then set $\mathcal{B}:=\mathcal{B}\cup\{\sigma.\varepsilon:C(x)\}$.
$\neg\{\}$-rule	If $\sigma.\varepsilon:(\neg\{u\})(x)\in\mathcal{B}$, and $x\neq u\notin\mathcal{B}_{Ind}^*$, then set $\mathcal{B}:=\mathcal{B}\cup\{x\neq u\}$.
$\{\}$-rule	If $\sigma.\varepsilon:\{u\}(x)\in\mathcal{B}$, and $x=u\notin\mathcal{B}_{Ind}^*$, then set $\mathcal{B}:=\mathcal{B}\cup\{x=u\}$.
$\neg@$-rule	If $\sigma.\varepsilon:(\neg@_uC)(x)\in\mathcal{B}$, and $\sigma.\varepsilon:(\neg C)(u)\notin\mathcal{B}$, then set $\mathcal{B}:=\mathcal{B}\cup\{\sigma.\varepsilon:(\neg C)(u)\}$.
$@$-rule	If $\sigma.\varepsilon:(@_uC)(x)\in\mathcal{B}$, and $\sigma.\varepsilon:C(u)\notin\mathcal{B}$, then set $\mathcal{B}:=\mathcal{B}\cup\{\sigma.\varepsilon:C(u)\}$.
$\neg\sqcup$-rule	If $\sigma.\varepsilon:(\neg(C_1\sqcup C_2))(x)\in\mathcal{B}$, and $\sigma.\varepsilon:(\neg C_1)(x)\notin\mathcal{B}$ or $\sigma.\varepsilon:(\neg C_2)(x)\notin\mathcal{B}$, then set $\mathcal{B}:=\mathcal{B}\cup\{\sigma.\varepsilon:(\neg C_1)(x),\sigma.\varepsilon:(\neg C_2)(x)\}$.
$\sqcup$-rule	If $\sigma.\varepsilon:(C_1\sqcup C_2)(x)\in\mathcal{B}$, and both $\sigma.\varepsilon:C_1(x)\notin\mathcal{B}$ and $\sigma.\varepsilon:C_2(x)\notin\mathcal{B}$, then set $\mathcal{B}:=\mathcal{B}\cup\{\sigma.\varepsilon:C(x)\}$ for some $C\in\{C_1,C_2\}$.
$\neg\forall$-rule	If $\sigma_0.\varepsilon_0:(\neg(\forall R.C))(x)\in\mathcal{B}$, and no individual name y exists with $\sigma_0.\varepsilon_0:R(x,y)\in\mathcal{B}$ and $\sigma_0.\varepsilon_0:(\neg C)(y)\in\mathcal{B}$, then introduce a new individual name y and set $\mathcal{B}:=\mathcal{B}\cup\{\sigma_0.\varepsilon_0:R(x,y),\sigma_0.\varepsilon_0:(\neg C)(y)\}$.
$\forall$-rule	If $\sigma_0.\varepsilon_0:(\forall R.C)(x)\in\mathcal{B}$, $\sigma_0.\varepsilon_0:R(x,y)\in\mathcal{B}$ and $\sigma_0.\varepsilon_0:C(y)\notin\mathcal{B}$, then set $\mathcal{B}:=\mathcal{B}\cup\{\sigma_0.\varepsilon_0:C(y)\}$.
$\neg\leq$-rule	If $\sigma_0.\varepsilon_0:(\neg(\leq nR.C))(x)\in\mathcal{B}$, and no n+1 individual names $y_1,\dots,y_{n+1}$ exist with ①$[y_i]_{=\mathcal{B}}\neq[y_j]_{=\mathcal{B}}$ for $1\leq i<j\leq n+1$ and ②both $\sigma_0.\varepsilon_0:R(x,y_k)\in\mathcal{B}$ and $\sigma_0.\varepsilon_0:C(y_k)\in\mathcal{B}$ for $1\leq k\leq n+1$, then introduce n+1 new individual names $y_1,\dots,y_{n+1}$ and set $\mathcal{B}:=\mathcal{B}\cup\{\sigma_0.\varepsilon_0:R(x,y_k)\|1\leq k\leq n+1\}\cup\{\sigma_0.\varepsilon_0:C(y_k)\|1\leq k\leq n+1\}\cup\{y_i\neq y_j\|1\leq i<j\leq n+1\}$.
$\leq$-rule	If $\sigma_0.\varepsilon_0:(\leq nR.C)(x)\in\mathcal{B}$, and there are n+1 individual names $y_1,\dots,y_{n+1}$ with ①$[y_i]_{=\mathcal{B}}\neq[y_j]_{=\mathcal{B}}$ for $1\leq i<j\leq n+1$ and ②both $\sigma_0.\varepsilon_0:R(x,y_k)\in\mathcal{B}$ and $\sigma_0.\varepsilon_0:C(y_k)\in\mathcal{B}$ for $1\leq k\leq n+1$, then select any two individual names y_i, y_j and set $\mathcal{B}:=\mathcal{B}\cup\{y_i=y_j\}$.

© 中国科学院软件研究所 http://www.c-s-a.org.cn

Table 2 Tableau expansion rules on EDDL(ALCQO) formulas except the after assertion and throughout assertion

表 2 EDDL(ALCQO)中除动作结果断言和动作过程断言之外的关于公式的扩展规则

Name	Rule
$\neg_f$-rule	If $\sigma.\varepsilon{:}\neg(C(x))\in\mathcal{B}$, and $\sigma.\varepsilon{:}(\neg C)(x)\notin\mathcal{B}$, then set $\mathcal{B}{:=}\mathcal{B}\cup\{\sigma.\varepsilon{:}(\neg C)(x)\}$.
$\neg\neg_f$-rule	If $\sigma.\varepsilon{:}\neg(\neg\varphi)\in\mathcal{B}$, and $\sigma.\varepsilon{:}\varphi\notin\mathcal{B}$, then set $\mathcal{B}{:=}\mathcal{B}\cup\{\sigma.\varepsilon{:}\varphi\}$.
$\neg\vee$-rule	If $\sigma.\varepsilon{:}\neg(\varphi\vee\psi)\in\mathcal{B}$, and $\sigma.\varepsilon{:}\neg\varphi\notin\mathcal{B}$ or $\sigma.\varepsilon{:}\neg\psi\notin\mathcal{B}$, then set $\mathcal{B}{:=}\mathcal{B}\cup\{\sigma.\varepsilon{:}\neg\varphi,\sigma.\varepsilon{:}\neg\psi\}$.
$\vee$-rule	If $\sigma.\varepsilon{:}\varphi\vee\psi\in\mathcal{B}$, and both $\sigma.\varepsilon{:}\varphi\notin\mathcal{B}$ and $\sigma.\varepsilon{:}\psi\notin\mathcal{B}$, then set $\mathcal{B}{:=}\mathcal{B}\cup\{\sigma.\varepsilon{:}\Phi\}$ for some $\Phi\in\{\varphi,\psi\}$.

表 3 给出了关于动作结果断言的扩展规则.虽然 EDDL(ALCQO)中将动作解释为由轨迹组成的集合,但实际上,动作结果断言中仅关注于动作轨迹的第 1 个和最后一个可能世界.因此,如果将形如$[\pi]\varphi$的公式改写为$\neg\langle\pi\rangle\neg\varphi$的形式,则可以将表 3 中的扩展规则一一映射为文献[9]中关于动作存在性断言$\langle\pi\rangle\varphi$的扩展规则.

Table 3 Tableau expansion rules on after assertions of EDDL(ALCQO)

表 3 EDDL(ALCQO)中关于动作结果断言的扩展规则

Name	Rule
$\neg;_{[\,]}$-rule	If $\sigma.\varepsilon{:}\neg[\pi_1;\pi_2]\varphi\in\mathcal{B}$, and $\sigma.\varepsilon{:}\neg[\pi_1][\pi_2]\varphi\notin\mathcal{B}$, then set $\mathcal{B}{:=}\mathcal{B}\cup\{\sigma.\varepsilon{:}\neg[\pi_1][\pi_2]\varphi\}$.
$;_{[\,]}$-rule	If $\sigma.\varepsilon{:}[\pi_1;\pi_2]\varphi\in\mathcal{B}$, and $\sigma.\varepsilon{:}[\pi_1][\pi_2]\varphi\notin\mathcal{B}$, then set $\mathcal{B}{:=}\mathcal{B}\cup\{\sigma.\varepsilon{:}[\pi_1][\pi_2]\varphi\}$.
$\neg?_{[\,]}$-rule	If $\sigma.\varepsilon{:}\neg[\psi?]\varphi\in\mathcal{B}$, and $\sigma.\varepsilon{:}\psi\notin\mathcal{B}$ or $\sigma.\varepsilon{:}\neg\varphi\notin\mathcal{B}$, then set $\mathcal{B}{:=}\mathcal{B}\cup\{\sigma.\varepsilon{:}\psi,\sigma.\varepsilon{:}\neg\varphi\}$.
$?_{[\,]}$-rule	If $\sigma.\varepsilon{:}[\psi?]\varphi\in\mathcal{B}$, and both $\sigma.\varepsilon{:}\neg\psi\notin\mathcal{B}$ and $\sigma.\varepsilon{:}\varphi\notin\mathcal{B}$, then set $\mathcal{B}{:=}\mathcal{B}\cup\{\sigma.\varepsilon{:}\neg\psi\}$ or set $\mathcal{B}{:=}\mathcal{B}\cup\{\sigma.\varepsilon{:}\varphi\}$.
$\neg\cup_{[\,]}$-rule	If $\sigma.\varepsilon{:}\neg[\pi_1\cup\pi_2]\varphi\in\mathcal{B}$, and both $\sigma.\varepsilon{:}\neg[\pi_1]\varphi\notin\mathcal{B}$ and $\sigma.\varepsilon{:}\neg[\pi_2]\varphi\notin\mathcal{B}$, then set $\mathcal{B}{:=}\mathcal{B}\cup\{\sigma.\varepsilon{:}\neg[\pi]\varphi\}$ for some $\pi\in\{\pi_1,\pi_2\}$.
$\cup_{[\,]}$-rule	If $\sigma.\varepsilon{:}[\pi_1\cup\pi_2]\varphi\in\mathcal{B}$, and $\sigma.\varepsilon{:}[\pi_1]\varphi\notin\mathcal{B}$ or $\sigma.\varepsilon{:}[\pi_2]\varphi\notin\mathcal{B}$, then set $\mathcal{B}{:=}\mathcal{B}\cup\{\sigma.\varepsilon{:}[\pi_1]\varphi,\sigma.\varepsilon{:}[\pi_2]\varphi\}$.
$\neg^*_{[\,]}$-rule	If $\sigma.\varepsilon{:}\neg[\pi^*]\varphi\in\mathcal{B}$, and there is no flag $X\equiv\neg[\pi^*]\varphi$ with $\sigma.\varepsilon{:}X\in\mathcal{B}$, then introduce a new string X and set $\mathcal{B}{:=}\mathcal{B}\cup\{X\equiv\neg[\pi^*]\varphi,\sigma.\varepsilon{:}X\}$.
X-rule	If there is some flag $X\equiv\neg[\pi^*]\varphi$ in $\mathcal{B}$ with $\sigma.\varepsilon{:}X\in\mathcal{B}$, $\sigma.\varepsilon{:}\neg\varphi\notin\mathcal{B}$ and $\sigma.\varepsilon{:}\neg[\pi]\neg X\notin\mathcal{B}$, then set $\mathcal{B}{:=}\mathcal{B}\cup\{\sigma.\varepsilon{:}\neg\varphi\}$ or set $\mathcal{B}{:=}\mathcal{B}\cup\{\sigma.\varepsilon{:}\varphi,\sigma.\varepsilon{:}\neg[\pi]\neg X\}$.
$^*_{[\,]}$-rule	If $\sigma.\varepsilon{:}[\pi^*]\varphi\in\mathcal{B}$, and $\sigma.\varepsilon{:}\varphi\notin\mathcal{B}$ or $\sigma.\varepsilon{:}[\pi][\pi^*]\varphi\notin\mathcal{B}$, then set $\mathcal{B}{:=}\mathcal{B}\cup\{\sigma.\varepsilon{:}\varphi,\sigma.\varepsilon{:}[\pi][\pi^*]\varphi\}$.
$\neg\mathrm{atom}_{[\,]}$-rule	If $\sigma.\varepsilon{:}\neg[(P,E)]\varphi\in\mathcal{B}$, and either $\{\sigma.\varepsilon{:}\psi\mid\psi\in P\}\not\subset\mathcal{B}$ or there is no prefix $\sigma'.\varepsilon'$ with $\varepsilon'=(\varepsilon\backslash E^{\neg})\cup E$ and $\sigma'.\varepsilon'{:}\neg\varphi\in\mathcal{B}$, then: (Step 1) if $\{\sigma.\varepsilon{:}\psi\mid\psi\in P\}\not\subset\mathcal{B}$ then set $\mathcal{B}{:=}\mathcal{B}\cup\{\sigma.\varepsilon{:}\psi\mid\psi\in P\}$, and (Step 2) if there is no prefix $\sigma'.\varepsilon'$ with $\varepsilon'=(\varepsilon\backslash E^{\neg})\cup E$ and $\sigma'.\varepsilon'{:}\neg\varphi\in\mathcal{B}$, then: if there is no preix $\sigma'.\varepsilon'$ with $\varepsilon'=(\varepsilon\backslash E^{\neg})\cup E$, then introduce a prefix $\sigma'.\varepsilon'{:=}\sigma;(P,E).(\varepsilon\backslash E^{\neg})\cup E$ and set $\mathcal{B}{:=}\mathcal{B}\cup\{\sigma'.\varepsilon'{:}\neg\varphi\}$, else set $\mathcal{B}{:=}\mathcal{B}\cup\{\sigma'.\varepsilon'{:}\neg\varphi\}$ for some prefix $\sigma'.\varepsilon'$ which is occurring in $\mathcal{B}$ with $\varepsilon'=(\varepsilon\backslash E^{\neg})\cup E$.
$\mathrm{atom}_{[\,]}$-rule	If $\sigma.\varepsilon{:}[(P,E)]\varphi\in\mathcal{B}$, $\{\sigma.\varepsilon{:}\neg\psi\mid\psi\in P\}\cap\mathcal{B}=\varnothing$, and some prefix $\sigma'.\varepsilon'$ exists with $\varepsilon'=(\varepsilon\backslash E^{\neg})\cup E$ and $\sigma'.\varepsilon'{:}\varphi\notin\mathcal{B}$, then set $\mathcal{B}{:=}\mathcal{B}\cup\{\sigma'.\varepsilon'{:}\varphi\}$ or set $\mathcal{B}{:=}\mathcal{B}\cup\{\sigma.\varepsilon{:}\neg\psi\}$ for some $\psi\in P$.

表 4 给出了关于动作过程断言的扩展规则.这些扩展规则比较直观地体现了EDDL(ALCQO)中对动作和动作过程断言的语义定义.其中,$\neg;_{\{\}}$-,$;_{\{\}}$-,$\neg^*_{\{\}}$-以及$^*_{\{\}}$-规则直观地体现了动作过程断言与动作结果断言在语义上的联系.$\neg\mathrm{atom}_{\{\}}$-和$\mathrm{atom}_{\{\}}$-规则看起来较为复杂,应用这两条规则后分别保证了下面两条性质:

(1) 如果$\sigma.\varepsilon{:}\neg\{(P,E)\}\varphi\in\mathcal{B}$,则:要么$\sigma.\varepsilon{:}\neg\varphi\in\mathcal{B}$,要么$\{\sigma.\varepsilon{:}\psi\mid\psi\in P\}\subseteq\mathcal{B}$并且存在某个前缀$\sigma'.\varepsilon'$,使得$\varepsilon'=(\varepsilon\backslash E^{\neg})\cup E$和$\sigma'.\varepsilon'{:}\neg\varphi\in\mathcal{B}$;

(2) 如果$\sigma.\varepsilon{:}\{(P,E)\}\varphi\in\mathcal{B}$,则必然有$\sigma.\varepsilon{:}\varphi\in\mathcal{B}$,并且:要么$\{\sigma.\varepsilon{:}\neg\psi\mid\psi\in P\}\cap\mathcal{B}\neq\varnothing$,要么对于满足$\varepsilon'=(\varepsilon\backslash E^{\neg})\cup E$的任一前缀$\sigma'.\varepsilon'$都有$\sigma'.\varepsilon'{:}\varphi\in\mathcal{B}$.

表 5 的扩展规则体现了本文在定义原子动作定义式的语义时采用的可能模型途径,与文献[9]中相应的扩展规则相同.其中,B_2-规则中对概念$D^{Regress(\sigma.\varepsilon)}$的递归构造过程与文献[9]中关于动态描述逻辑D-ALCQO的相应的递归构造过程相同,目的是保证如下性质:对于任一模型$M=(\Sigma,\Delta_M,I)$(其中,$\Sigma=(W,\cdot^T)$)以及该模型中的任意两个可能世界w,w':如果$(w,w')\in\sigma^T$,则必然有$(D^{Regress(\sigma.\varepsilon)})^{I(w)}=D^{I(w')}$.对该性质的证明可参见文献[9].

最后可以给出 EDDL(ALCQO)的表判定算法.

算法 1(判定 EDDL(ALCQO)中公式的可满足性). 按以下步骤判定公式φ相对于$\mathcal{T}$和$\mathcal{A}c$是否可满足:

(1) 构造分枝$\mathcal{B}{:=}\{\sigma_0.\varepsilon_0{:}nf_{\mathcal{T},\mathcal{A}c}(\varphi)\}$;

(2) 以任意顺序应用表 1~表 5 的扩展规则对 $\mathcal{B}$ 进行扩展;

www.jos.org.cn

(3) 如果通过扩展后能得到某个饱和的分枝 $\mathcal{B}'$,并且 $\mathcal{B}'$是无冲突的以及不可忽略的,则返回结果“φ相对于 $\mathcal{T}$和 $\mathcal{A}c$ 是可满足的”;否则,返回结果“φ相对于 $\mathcal{T}$和 $\mathcal{A}c$ 是不可满足的”.

Table 4 Tableau expansion rules on throughout assertions of EDDL(ALCQO)

表 4 EDDL(ALCQO)中关于动作过程断言的扩展规则

Name	Rule
$\neg;_{\{\}}$-rule	If $\sigma.\mathcal{E}{:}\neg\{\pi_1;\pi_2\}\varphi\in\mathcal{B}$, and both $\sigma.\mathcal{E}{:}\neg\{\pi_1\}\varphi\notin\mathcal{B}$ and $\sigma.\mathcal{E}{:}\neg[\pi_1]\{\pi_2\}\varphi\notin\mathcal{B}$, then set $\mathcal{B}:=\mathcal{B}\cup\{\sigma.\mathcal{E}{:}\neg\{\pi_1\}\varphi\}$ or set $\mathcal{B}:=\mathcal{B}\cup\{\sigma.\mathcal{E}{:}\neg[\pi_1]\{\pi_2\}\varphi\}$.
$;_{\{\}}$-rule	If $\sigma.\mathcal{E}{:}\{\pi_1;\pi_2\}\varphi\in\mathcal{B}$, and $\sigma.\mathcal{E}{:}\{\pi_1\}\varphi\notin\mathcal{B}$ or $\sigma.\mathcal{E}{:}[\pi_1]\{\pi_2\}\varphi\notin\mathcal{B}$, then set $\mathcal{B}:=\mathcal{B}\cup\{\sigma.\mathcal{E}{:}\{\pi_1\}\varphi, \sigma.\mathcal{E}{:}[\pi_1]\{\pi_2\}\varphi\}$.
$\neg?_{\{\}}$-rule	If $\sigma.\mathcal{E}{:}\neg\{\psi?\}\varphi\in\mathcal{B}$, and $\sigma.\mathcal{E}{:}\psi\notin\mathcal{B}$ or $\sigma.\mathcal{E}{:}\neg\varphi\notin\mathcal{B}$, then set $\mathcal{B}:=\mathcal{B}\cup\{\sigma.\mathcal{E}{:}\psi, \sigma.\mathcal{E}{:}\neg\varphi\}$.
$?_{\{\}}$-rule	If $\sigma.\mathcal{E}{:}\{\psi?\}\varphi\in\mathcal{B}$, and both $\sigma.\mathcal{E}{:}\neg\psi\notin\mathcal{B}$ and $\sigma.\mathcal{E}{:}\varphi\notin\mathcal{B}$, then set $\mathcal{B}:=\mathcal{B}\cup\{\sigma.\mathcal{E}{:}\neg\psi\}$ or set $\mathcal{B}:=\mathcal{B}\cup\{\sigma.\mathcal{E}{:}\varphi\}$.
$\neg\cup_{\{\}}$-rule	If $\sigma.\mathcal{E}{:}\neg\{\pi_1\cup\pi_2\}\varphi\in\mathcal{B}$, and both $\sigma.\mathcal{E}{:}\neg\{\pi_1\}\varphi\notin\mathcal{B}$ and $\sigma.\mathcal{E}{:}\neg\{\pi_2\}\varphi\notin\mathcal{B}$, then set $\mathcal{B}:=\mathcal{B}\cup\{\sigma.\mathcal{E}{:}\neg\{\pi_1\}\varphi\}$ or set $\mathcal{B}:=\mathcal{B}\cup\{\sigma.\mathcal{E}{:}\neg\{\pi_2\}\varphi\}$.
$\cup_{\{\}}$-rule	If $\sigma.\mathcal{E}{:}\{\pi_1\cup\pi_2\}\varphi\in\mathcal{B}$, and $\sigma.\mathcal{E}{:}\{\pi_1\}\varphi\notin\mathcal{B}$ or $\sigma.\mathcal{E}{:}\{\pi_2\}\varphi\notin\mathcal{B}$, then set $\mathcal{B}:=\mathcal{B}\cup\{\sigma.\mathcal{E}{:}\{\pi_1\}\varphi, \sigma.\mathcal{E}{:}\{\pi_2\}\varphi\}$.
$\neg^*_{\{\}}$-rule	If $\sigma.\mathcal{E}{:}\neg\{\pi^*\}\varphi\in\mathcal{B}$, and $\sigma.\mathcal{E}{:}\neg[\pi^*]\{\pi\}\varphi\notin\mathcal{B}$, then set $\mathcal{B}:=\mathcal{B}\cup\{\sigma.\mathcal{E}{:}\neg[\pi^*]\{\pi\}\varphi\}$.
$^*_{\{\}}$-rule	If $\sigma.\mathcal{E}{:}\{\pi^*\}\varphi\in\mathcal{B}$, and $\sigma.\mathcal{E}{:}[\pi^*]\{\pi\}\varphi\notin\mathcal{B}$, then set $\mathcal{B}:=\mathcal{B}\cup\{\sigma.\mathcal{E}{:}[\pi^*]\{\pi\}\varphi\}$.
$\neg$atom$_{\{\}}$-rule	If $\sigma.\mathcal{E}{:}\neg\{(P,E)\}\varphi\in\mathcal{B}$, and either $\sigma.\mathcal{E}{:}\neg\varphi\notin\mathcal{B}$ or both $\{\sigma.\mathcal{E}{:}\psi \mid \psi\in P\}\not\subset\mathcal{B}$ and no prefix $\sigma'.\mathcal{E}'$ exists with $\mathcal{E}'=(\mathcal{E}\backslash E^{\neg})\cup E$ and $\sigma'.\mathcal{E}'{:}\neg\varphi\in\mathcal{B}$, then set $\mathcal{B}:=\mathcal{B}\cup\{\sigma.\mathcal{E}{:}\neg\varphi\}$, or do the following operations in the case that no prefix $\sigma'.\mathcal{E}'$ exists with $\mathcal{E}'=(\mathcal{E}\backslash E^{\neg})\cup E$ and $\sigma'.\mathcal{E}'{:}\neg\varphi\in\mathcal{B}$: if there is some prefix $\sigma'.\mathcal{E}'$ occurring in $\mathcal{B}$ with $\mathcal{E}'=(\mathcal{E}\backslash E^{\neg})\cup E$, then set $\mathcal{B}:=\mathcal{B}\cup\{\sigma.\mathcal{E}{:}\psi \mid \psi\in P\}\cup\{\sigma'.\mathcal{E}'{:}\neg\varphi\}$, else introduce a prefix $\sigma'.\mathcal{E}':=\sigma;(P,E).(\mathcal{E}\backslash E^{\neg})\cup E$ and set $\mathcal{B}:=\mathcal{B}\cup\{\sigma.\mathcal{E}{:}\psi \mid \psi\in P\}\cup\{\sigma'.\mathcal{E}'{:}\neg\varphi\}$.
atom$_{\{\}}$-rule	If $\sigma.\mathcal{E}{:}\{(P,E)\}\varphi\in\mathcal{B}$, and either $\sigma.\mathcal{E}{:}\varphi\notin\mathcal{B}$ or both $\{\sigma.\mathcal{E}{:}\neg\psi \mid \psi\in P\}\cap\mathcal{B}=\varnothing$ and there is some prefix $\sigma'.\mathcal{E}'$ with $\mathcal{E}'=(\mathcal{E}\backslash E^{\neg})\cup E$ and $\sigma'.\mathcal{E}'{:}\varphi\notin\mathcal{B}$, then: (Step 1) if $\sigma.\mathcal{E}{:}\varphi\notin\mathcal{B}$ then set $\mathcal{B}:=\mathcal{B}\cup\{\sigma.\mathcal{E}{:}\varphi\}$; (Step 2) if $\{\sigma.\mathcal{E}{:}\neg\psi \mid \psi\in P\}\cap\mathcal{B}=\varnothing$ and there is some prefix $\sigma'.\mathcal{E}'$ with $\mathcal{E}'=(\mathcal{E}\backslash E^{\neg})\cup E$ and $\sigma'.\mathcal{E}'{:}\varphi\notin\mathcal{B}$, then set $\mathcal{B}:=\mathcal{B}\cup\{\sigma'.\mathcal{E}'{:}\varphi\}$ or set $\mathcal{B}:=\mathcal{B}\cup\{\sigma.\mathcal{E}{:}\neg\psi\}$ for some $\psi\in P$.

Table 5 Tableau expansion rules on effects of EDDL(ALCQO) actions

表 5 EDDL(ALCQO)中关于动作效果的扩展规则

Name	Rule
B_1-rule	If $\sigma.\mathcal{E}{:}\varphi\in\mathcal{B}$, φ is a simple formula, $\varphi\notin\mathcal{E}$, and $\sigma_0.\mathcal{E}_0{:}\varphi\notin\mathcal{B}$, then set $\mathcal{B}:=\mathcal{B}\cup\{\sigma_0.\mathcal{E}_0{:}\varphi\}$.
B_2-rule	If $\sigma.\mathcal{E}{:}D(u)\in\mathcal{B}$, D is some concept with the form $\forall R.C$ or $\neg\forall R.C$, and $\sigma_0.\mathcal{E}_0{:}D^{Regress(\sigma.\mathcal{E})}(u)\notin\mathcal{B}$, then set $\mathcal{B}:=\mathcal{B}\cup\{\sigma_0.\mathcal{E}_0{:}D^{Regress(\sigma.\mathcal{E})}(u)\}$.

3 判定算法的性质

本节证明算法 1 是可终止的、可靠的和完备的.基本思路与文献[9]中对 D-ALCQO 判定算法的证明相似.

首先,对于EDDL(ALCQO)中的任一概念、公式或者动作Y,用$size(Y)$表示如下集合中的元素在Y中累计出现的次数:$N_C\cup N_R\cup N_I\cup\{@,\neg,\sqcup,\forall,\vee,[,],\{,\},?,\cup,;,^*\}$.此外,对于EDDL(ALCQO)中的任一公式$\varphi$,用$cl_f(\varphi)$表示满足以下条件的最小集合:

(1) $\varphi\in cl_f(\varphi)$;

(2) 如果$\neg\psi\in cl_f(\varphi)$,则$\psi\in cl_f(\varphi)$;如果$\psi\in cl_f(\varphi)$并且$\psi$不是以“$\neg$”开头的公式,则$\neg\psi\in cl_f(\varphi)$;

(3) 如果$\psi_1\vee\psi_2\in cl_f(\varphi)$,$[\psi_1?]\psi_2\in cl_f(\varphi)$或者$\{\psi_1?\}\psi_2\in cl_f(\varphi)$,则$\{\psi_1,\psi_2\}\subseteq cl_f(\varphi)$;

(4) 如果$[(P,E)]\psi\in cl_f(\varphi)$或者$\{(P,E)\}\psi\in cl_f(\varphi)$,则$P\cup E\cup\{\psi\}\subseteq cl_f(\varphi)$;

(5) 如果$[\pi\cup\pi']\psi\in cl_f(\varphi)$,则$\{[\pi]\psi,[\pi']\psi\}\subseteq cl_f(\varphi)$;如果$\{\pi\cup\pi'\}\psi\in cl_f(\varphi)$,则$\{\{\pi\}\psi,\{\pi'\}\psi\}\subseteq cl_f(\varphi)$;

(6) 如果$[\pi;\pi']\psi\in cl_f(\varphi)$,则$[\pi][\pi']\psi\in cl_f(\varphi)$;如果$\{\pi;\pi'\}\psi\in cl_f(\varphi)$,则$\{\pi\}\psi\in cl_f(\varphi)$以及$[\pi]\{\pi'\}\psi\in cl_f(\varphi)$;

(7) 如果$[\pi^*]\psi\in cl_f(\varphi)$,则$[\pi][\pi^*]\psi\in cl_f(\varphi)$;如果$\{\pi^*\}\psi\in cl_f(\varphi)$,则$[\pi^*]\{\pi\}\psi\in cl_f(\varphi)$;

(8) 如果$\neg(C(x))\in cl_f(\varphi)$,则$(\neg C)(x)\in cl_f(\varphi)$;

(9) 如果$(\neg C)(x)\in cl_f(\varphi)$,则$C(x)\in cl_f(\varphi)$;如果$C(x)\in cl_f(\varphi)$并且$C$不是以“$\neg$”开头的概念,则$(\neg C)(x)\in cl_f(\varphi)$;

(10) 如果$@_uC(x)\in cl_f(\varphi)$,则$C(u)\in cl_f(\varphi)$;如果$(C\sqcup D)(x)\in cl_f(\varphi)$,则$\{C(x),D(x)\}\subseteq cl_f(\varphi)$.

© 中国科学院软件研究所 http://www.c-s-a.org.cn

显然,集合$cl_f(nf_{\mathcal{T},\mathcal{A}c}(\varphi))$中元素的个数$\#cl_f(nf_{\mathcal{T},\mathcal{A}c}(\varphi))$与$size(nf_{\mathcal{T},\mathcal{A}c}(\varphi))$成线性关系.

此外,对于 EDDL(ALCQO)中的任一公式φ,用 $AtomAct(\varphi)$表示由出现在φ中的所有原子动作组成的集合;在此基础上,用 $Eff(\varphi)$表示集合 $\bigcup_{(P_i,E_i)\in AtomAct(\varphi)} E_i$.

接下来,令s:=$size(nf_{\mathcal{T},\mathcal{A}c}(\varphi))$,$f$:=$\#cl_f(nf_{\mathcal{T},\mathcal{A}c}(\varphi))$,$e$:=$\#Eff(nf_{\mathcal{T},\mathcal{A}c}(\varphi))$,$count_I$为$nf_{\mathcal{T},\mathcal{A}c}(\varphi)$中出现的不同的个体名的个数,$count_\forall$为概念构造符$\forall$在$nf_{\mathcal{T},\mathcal{A}c}(\varphi)$中累计出现的次数,$count_*$为动作构造符*在$nf_{\mathcal{T},\mathcal{A}c}(\varphi)$中累计出现的次数.参照文献[9]中对判定算法可终止性的证明,可类似地得出算法 1 的以下性质:

(1) 每应用一次任意一条扩展规则之后,分枝 $\mathcal{B}$ 的元素个数都至少增加 1;

(2) 所有可能性标记都是通过应用$\neg^*{}_{[\,]}$-规则引入的;$\mathcal{B}$中出现的所有可能性标记最多为$count_*$个;

(3) 除初始前缀$\sigma_0.\varepsilon_0$之外,所有前缀都是通过应用$\neg$atom$_{[\,]}$-规则和$\neg$atom$_{\{\}}$-规则引入的,因而最多引入 2^e-1 个前缀;

(4) 对于任一非初始前缀$\sigma.\varepsilon$,分枝 $\mathcal{B}$ 中出现的以$\sigma.\varepsilon$为前缀的带前缀的公式最多有 f 个;

(5) 由于$\neg\forall$-规则的应用,在分枝$\mathcal{B}$中最多会新引入$NUM_\forall$个个体名,其中,

$$NUM_\forall=(count_I\times(count_I-1)+2)^{count_\forall}\times(count_\forall\times(2^e-1))+count_\forall;$$

(6) 在没有应用$\neg\forall$-规则的情况下,分枝$\mathcal{B}$中出现的以$\sigma_0.\varepsilon_0$为前缀的带前缀的公式最多为$f+2^{p_1(s)}\times p_2(e)\times count_\forall\times(2^e-1)$个,其中,$p_1$和$p_2$为多项式;由于应用$\neg\forall$-规则引入新的个体名,进而应用其他扩展规则后,最多将增加$(2\times NUM_\forall)+NUM_\forall\times(NUM_\forall-1)\times(k\times(s+2^{p_1(s)}\times p_{2_1}(e)))$个以$\sigma_0.\varepsilon_0$为前缀的带前缀的公式,其中,$k$为某个自然数;

(7) 分枝$\mathcal{B}$中个体名的数量最多为$count_I+NUM_\forall$.因此,$\mathcal{B}$中关于个体名的等式最多为$count_I+NUM_\forall-1$ 个,关于个体名的不等式最多为$(count_I+NUM_\forall)\times(count_I+NUM_\forall-1)\times1/2$.

根据上面的性质(2)~性质(7),集合$\mathcal{B}$的元素个数是有限的,并且最多为$p_3(s)\times 2^{p_4(s)}\times count_I^{4\times count_\forall}$ 个,其中的p_3,p_4为多项式;再结合性质(1),可得出算法 1 是可终止的,即如下定理成立.

定理 1. 算法 1 是可终止的.

为了证明算法的可靠性和完备性,关键是引入分枝-模型映射,将分枝中出现的前缀与模型中的可能世界一一对应起来,然后再引入分枝的可满足性.

定义 19(分枝-模型映射). 令Σ_{prefix}是由出现在$\mathcal{B}$中的所有前缀组成的集合,$M=(\Sigma,\Delta_M,I)$为EDDL(ALCQO)模型,其中,$\Sigma=(W,\cdot^T)$;将函数$\delta:\Sigma_{prefix}\rightarrow W$称为从$\mathcal{B}$到$M$的分枝-模型映射,当且仅当对于$\Sigma_{prefix}$中的任意两个前缀$\sigma.\varepsilon,\sigma'.\varepsilon'$来说:如果存在原子动作$(P,E)$使得$\sigma'=\sigma;(P,E)$和$\varepsilon'=(\varepsilon\backslash E^\neg)\cup E$,则必然有:

(1) 对于出现在$\mathcal{B}$中的任一简单概念名C,都有$C^{I(\delta(\sigma'.\varepsilon'))}=(C^{I(\delta(\sigma.\varepsilon))}\cup\{u^I|C(u)\in E\})\backslash\{u^I|\neg C(u)\in E\}$;

(2) 对于出现在$\mathcal{B}$中的任一角色名R,都有$R^{I(\delta(\sigma'.\varepsilon'))}=(R^{I(\delta(\sigma.\varepsilon))}\cup\{(u^I,v^I)|R(u,v)\in E\})\backslash\{(u^I,v^I)|\neg R(u,v)\in E\}$.

定义 20(分枝的可满足性). 令$\mathcal{B}$为任一分枝,$M=(\Sigma,\Delta_M,I)$为EDDL(ALCQO)模型,δ为从$\mathcal{B}$到M的分枝-模型映射,则称模型M和映射δ满足分枝$\mathcal{B}$,表示为$M,\delta\models\mathcal{B}$,当且仅当:① 对于$\mathcal{B}$中任一带前缀的公式$\sigma.\varepsilon:\varphi$,都有$(M,\delta(\sigma.\varepsilon))\models\varphi$;② 对于$\mathcal{B}$中任一关于个体名的等式$u=v$,都有$u^I=v^I$;③ 对于$\mathcal{B}$中任一关于个体名的不等式$u\neq v$,都有$u^I\neq v^I$.

对于分枝 $\mathcal{B}$,若存在某个模型 M 以及相应的映射δ使得 $M,\delta\models\mathcal{B}$,则称 $\mathcal{B}$ 是可满足的,否则称 $\mathcal{B}$ 为不可满足的.

基于上述定义,类似于文献[9],可依次证明以下引理.

引理 1. 对于在分枝 $\mathcal{B}$ 上应用任意一条扩展规则的过程来说,$\mathcal{B}$ 在扩展之前是可满足的当且仅当可以在扩展后得到某个可满足的分枝.

引理 2. 如果分枝 $\mathcal{B}$ 是饱和的、无冲突的并且不可忽略的,则 $\mathcal{B}$ 一定是可满足的.

引理 3. 如果算法 1 构造的初始分枝$\mathcal{B}$:={$\sigma_0.\varepsilon_0$:$nf_{\mathcal{T},\mathcal{A}c}(\varphi)$}是可满足的,则在应用扩展规则后得到的所有饱和的并且可满足的分枝中,必然存在某个分枝是不可忽略的.

引理 4. 如果分枝 $\mathcal{B}$ 存在冲突,则 $\mathcal{B}$ 一定是不可满足的.

最后,根据定理 1 以及引理 1~引理 4,容易证明算法 1 的可靠性和完备性,即:

定理 2. 算法 1 返回结果"φ相对于 $\mathcal{T}$ 和 $\mathcal{A}c$ 是可满足的"当且仅当φ相对于 $\mathcal{T}$ 和 $\mathcal{A}c$ 是可满足的.

4 应用实例

EDDL($\mathcal{X}$)继承了动态描述逻辑中对动作的描述和推理能力.以基于描述逻辑 $\mathcal{X}$ 的领域知识作为背景,*EDDL*($\mathcal{X}$)可以从动作的前提条件以及执行动作之后产生的影响两个方面对原子动作进行刻画.在此基础上,应用测试、顺序、选择、迭代等构造符,可以刻画出具有复杂控制结构的动作.针对这些动作,类似于文献[10]给出的相关定理,同样可以将动作的可实现性、可执行性、投影、规划等推理问题转换为 *EDDL*($\mathcal{X}$)中公式的可满足性问题,然后基于开世界假设在信息不完全的情况下进行判定和推理.

此外,*EDDL*($\mathcal{X}$)还可以对动作的执行过程进行考察,判断在某个状态下执行某个动作的过程中能否使得某个公式一直成立.令 $\mathcal{T},\mathcal{A}c$ 分别为 TBox 和 ActBox,$\mathcal{A}$ 为公式集合,π为动作,φ为公式,并且π和φ中出现的各个原子动作都是相对于 $\mathcal{T}$ 和 $\mathcal{A}c$ 被定义的;基于 *EDDL*($\mathcal{X}$)的语义模型对该推理问题定义如下:

定义 21. 动作π在公式集$\mathcal{A}$刻画的状态上执行的过程中一直使得公式φ成立,当且仅当对于任一模型 $M=(\Sigma,\Delta_M,I)$(其中,$\Sigma=(W,\cdot^T)$)以及该模型中的任一可能世界 w,如果 $M\vDash\mathcal{T},M\vDash\mathcal{A}c$ 以及 $(M,w)\vDash\mathcal{A}$,则必然有 $(M,w)\vDash\{\pi\}\varphi$.

根据定义,可以将该推理问题转换为 *EDDL*($\mathcal{X}$)中公式的可满足性问题.即:

定理 3. 动作π在公式集 $\mathcal{A}$ 刻画的状态上执行的过程中一直使得公式φ成立,当且仅当公式$\neg(Conj(\mathcal{A})\rightarrow\{\pi\}\varphi)$相对于 $\mathcal{T}$ 和 $\mathcal{A}c$ 是不可满足的.其中的 $Conj(\mathcal{A})$表示由集合 $\mathcal{A}$ 的各个元素作为合取项构成的合取式.

EDDL($\mathcal{X}$)为语义 Web 服务的建模和推理提供了进一步的支持.例如,假设某个 Web 服务系统可以让持有信用卡的用户购买图书和 CD,下面应用 EDDL(ALCQO)对该系统进行建模和推理.

模型中用到以下概念名:*person*,*creditCard*,*cd*,*book*,*customer*,*valuableCust*,*instore*;用到以下角色名:*bought*, *has*.

TBox $\mathcal{T}$ 由以下概念定义式组成:

$$customer\equiv person\sqcap\exists has.creditCard$$

$$VIPCustomer\equiv customer\sqcap\geq 3\ bought.(cd\sqcup book)$$

上述定义式的直观含义为:顾客是持有信用卡的人员;VIP 顾客是购买了至少 3 本 CD 或图书的顾客.

基于上述领域知识,对该系统中提供的原子服务描述如下:

$$buyCD(u,v)\equiv(\{customer(u),cd(v),instore(v),\neg bought(u,v)\},\{\neg instore(v),bought(u,v)\})$$

$$buyBook(u,v)\equiv(\{customer(u),book(v),instore(v),\neg bought(u,v)\},\{\neg instore(v),bought(u,v)\})$$

$$order(v)\equiv(\{book(v)\vee cd(v),\neg instore(v)\},\{instore(v)\}).$$

其中,$buyCD(u,v)$和 $buyBook(u,v)$分别让顾客购买 CD 和购买图书,$order(v)$让管理员从出版商进购 CD 或图书.对 $buyCD(u,v)$的刻画表明:如果公式 $customer(u),cd(v),instore(v)$以及$\neg bought(u,v)$成立,则该服务可以被执行,并且在执行后产生的影响是使得公式$\neg instore(v)$和 $bought(u,v)$成立.

在上述原子服务的基础上,还可以提供组合服务 $buy(u,v)$,描述为

$$buy(u,v)\equiv buyCD(u,v)\cup buyBook(u,v).$$

另外,还可以针对 VIP 顾客提供组合服务 $VIPbuy(u,v)$,描述为

$$VIPbuy(u,v)\equiv VIPCustomer(u)?;((instore(v)?;buy(u,v))\cup(\neg instore(v)?;order(v);buy(u,v))).$$

该组合服务的直观含义为:首先判断公式 $VIPCustomer(u)$是否成立;在 $VIPCustomer(u)$成立的前提下,如果 $instore(v)$也成立,则执行组合服务 $buy(u,v)$,否则需要先执行原子服务 $order(v)$后再执行组合服务 $buy(u,v)$.

假设在当前状态下有以下公式成立:

$$VIPCustomer(Tom),person(Jack),creditCard(Visa),has(Jack,Visa),book(HarryPotter),$$
$$\neg instore(HarryPotter),\neg bought(Tom,HarryPotter),\neg bought(Jack,HarryPotter),$$

其中,*Tom*,*Jack*,*Visa* 和 *HarryPotter* 为个体名.在此基础上,可以基于上述模型对 Web 服务进行推理.

首先,类似于文献[10]给出的对动作可执行性的判定方式,可以推断出以下结论:服务 $buy(Jack,HarryPotter)$ 在当前状态下是不可执行的,而组合服务 $order(HarryPotter);buyBook(Jack,HarryPotter)$在当前状态下是可以执

行的,服务 *VIPbuy*(*Tom*,*HarryPotter*)在当前状态下也是可以执行的.

其次,类似于文献[10]给出的对投影问题的判定方式,可以推断出如下结论:在当前状态下执行 Web 服务 *VIPbuy*(*Tom*,*HarryPotter*)后,将使得公式 *bought*(*Tom*,*HarryPotter*)成立.

第三,类似于文献[10]给出的对规划问题的判定方式,可以证明组合服务 *order*(*HarryPotter*);*buyBook*(*Jack*,*HarryPotter*)是在当前状态下实现目标 *bought*(*Jack*,*HarryPotter*)的一个规划.

最后,还可以根据本文的定理 3 对动作的执行过程进行推理.例如,可以推理出如下结论:在当前状态下执行 Web 服务 *VIPbuy*(*Tom*,*HarryPotter*)的过程中,公式 *VIPCustomer*(*Tom*)将一直成立,但公式$\neg$*instore*(*HarryPotter*)则不然.

5 相关工作

语义Web服务的基本思路是将Web服务作为一种知识进行描述和推理,在此基础上实现Web服务的自动发现、组合和执行.Web服务与动作具有相似的动态特征;以基于逻辑的动作理论为基础,可以对语义Web服务进行建模和推理.目前,比较成熟的基于逻辑的动作理论可以分为两大类:一类是采用命题语言的动作理论,例如命题STRIPS系统[14]、基于命题动态逻辑的动作系统[15]等.与这类动作理论相比,*EDDL*($\mathcal{X}$)在描述逻辑$\mathcal{X}$的基础上对领域知识进行刻画,从而具有更强的描述能力.例如,本文例子中由TBox $\mathcal{T}$刻画的知识是命题STRIPS系统和命题动态逻辑等无法处理的.另一类是基于谓词逻辑的动作理论,例如情境演算[16]、流演算[17]等.这类系统具有很强的描述能力,但由于一阶谓词逻辑的不可判定性,使得相关推理受到很大限制,需要借助额外的定理证明工具.例如,本文例子也可以转换为情境演算中的表示形式,但相应的推理需要借助定理证明工具来完成.

采用命题语言的动作理论与基于谓词逻辑的动作理论之间存在着一个关于描述能力和推理能力的鸿沟,将描述逻辑与动作理论结合之后有望填补这个鸿沟.基于这种观点,其他研究者在最近几年的国际会议上发表了相关研究成果.Baader等人[18]首先构建了一个基于描述逻辑的动作形式系统.在该系统中,由概念定义式组成的TBox被用于刻画动作理论的背景信息;由个体断言组成的ABox被用于刻画世界状态;每个原子动作被刻画为(*pre*,*occ*, *post*)的三元组形式,分别表示执行动作之前必须满足的前提条件、执行过程中的不确定成分以及执行后产生的影响,其中的*pre*,*occ*和*post*都用描述逻辑中的个体断言来刻画,并且*occ*和*post*中出现的必须是相对于TBox的简单公式.原子动作的有限序列构成复杂动作.在此基础上,该文将动作的可执行性问题和投影问题转换为描述逻辑中ABox的一致性问题来解决.Baader等人对该动作系统进行了许多限制,不允许测试、选择、迭代等具有复杂结构的动作.Gu和Soutchanski[19]以描述逻辑为参照,进一步提出对情景演算的一种改造形式.一方面,在基于情景演算的动作理论的基础上,将刻画动作以及初始情景的公式都限制为C^2逻辑中的公式;另一方面,将描述逻辑中的TBox和RBox引入到情景演算中.最终得到的动作系统既具有较强的描述能力,又保证了相关推理问题是可判定的.在此基础上,该文也对动作的可执行性问题和投影问题进行了深入考察.

上述两项工作的基本思路都是从情景演算出发,以描述逻辑的研究成果作为参照,对情景演算加以限定.使得最终得到的动作系统在具有较强描述能力的同时又保证了相关推理问题的可判定性.因此,他们给出的动作系统本质上都是情景演算的特殊形式.这种途径使得关于情景演算的许多研究成果和工具都可以作为参考或直接使用,但另一方面也不可避免地继承了情景演算的某些局限.例如,相关推理问题仍然要转换为一阶逻辑的定理证明问题.此外,情景演算中应用 GOLOG 或直接应用 Prolog 进行规划时,Prolog 所基于的闭世界假设要求所输入的信息是完全的,在信息不完全的情况下需要借助定理证明器来进行额外的处理.

与上述两项工作相比,本文是从描述逻辑出发,将动作及动作理论嵌入到描述逻辑中,构建出一类逻辑系统,即扩展的动态描述逻辑 *EDDL*($\mathcal{X}$).在对动作的刻画方面,除了原子动作和顺序动作之外,*EDDL*($\mathcal{X}$)中还引入了测试、选择、迭代等动作构造符,从而可以描述 if φ then π else π'以及 while φ do π等复杂的控制结构.同时,*EDDL*($\mathcal{X}$)引入了形如$\{a\}\varphi$和$[a]\varphi$的公式,从而可以从动作的执行过程和执行结果两个方面对关于动作的更多类型的推理问题进行直观刻画.在对动作的推理方面,可以将这些推理问题转换为公式的可满足性问题;以基于开世界假设的判定算法为基础,可以在信息不完全的情况下进行判定和推理.最后,*EDDL*($\mathcal{X}$)还提供了一类逻辑系

© 中国科学院软件研究所 http://www.c-s-a.org.cn

统.根据使用者对描述能力的需求,可以将其中的 X 替换为不同的描述逻辑,相应地可以获得不同的推理性能.例如在本文的例子中,对组合服务 $buy(u,v)$和 $VIPbuy(u,v)$的刻画是文献[18]给出的系统无法胜任的.在信息不完全的情况下对这些服务的各种推理,尤其是对动作执行过程的推理,则是文献[19]给出的系统所不能胜任的.

最后,在对动作执行过程的刻画和推理方面,本文主要参考了Pratt关于命题动态逻辑的相关工作.Pratt[12]首次在命题动态逻辑PDL(propositional dynamic logic)中引入了关于动作执行过程的模态词,用以构造形如$\{\pi\}\varphi$的动作过程断言,并给出了相应的表判定算法.本文将该研究成果与描述逻辑以及基于可能模型途径的动作理论有机地结合起来,得到一类扩展的动态描述逻辑$EDDL(X)$.其中,EDDL(ALCQO)的表判定算法可以看作是将描述逻辑ALCQO的表判定算法、Pratt给出的关于命题动态逻辑的判定算法以及对可能模型途径的处理有机结合后的产物.

6 结 论

本文给出了一类扩展的动态描述逻辑 $EDDL(X)$.以 X 为描述逻辑 ALCQO 的情况为例,本文给出了 EDDL (ALCQO)的表判定算法.$EDDL(X)$将动态逻辑、基于可能模型途径的动作理论以及由 X 代表的描述逻辑有机地结合起来,在一个逻辑系统内对基于描述逻辑的领域知识以及关于动作的知识进行统一的描述和推理.$EDDL(X)$既具有较强的对动作的描述能力,又提供了丰富的推理功能.以公式的可满足性问题为基础,可以分别从动作执行过程和动作执行结果两个方面对动作进行较为全面的推理和考察.

针对语义 Web 服务的自动发现和自动组合问题,基本途径是应用某种形式系统对语义 Web 服务进行建模和推理,在此基础上实现自动发现和自动组合.形式系统的描述能力与推理能力之间是一对矛盾关系,除了寻求两者之间的折衷之外,一个现实的途径是提供具有不同描述能力(相应地具有不同的推理能力)的多套工具,让使用者按需使用.$EDDL(X)$与动态描述逻辑一起,为语义 Web 服务的建模和推理提供了一种有效的途径和工具.下一步的工作是将它们进一步应用于语义 Web 服务的自动发现和自动组合.

致谢 感谢审稿专家提出的宝贵意见.

References:

[1] Horrocks I, Patel-Schneider PF, Harmelen FV. From SHIQ and RDF to OWL: The making of a Web ontology language. Journal of Web Semantics, 2003,1(1):7−26.

[2] Bonatti P, Lutz C, Wolter F. Description logics with circumscription. In: Doherty P, Mylopoulos J, Welty C, eds. Proc. of the 10th Int'l Conf. on Principles of Knowledge Representation and Reasoning. Menlo Park: AAAI Press, 2006. 400−410.

[3] Ma Y, Hitzler P, Lin ZQ. Algorithms for paraconsistent reasoning with OWL. In: Franconi E, Kifer M, May W, eds. Proc. of the 4th European Semantic Web Conf. Berlin: Springer-Verlag, 2007. 399−413.

[4] Li YH, Xu BW, Lu JJ, Kang DZ. On computational complexity of the extended fuzzy description logic wirh numerical restriction. Journal of Software, 2006,17(5):968−975 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/17/968.htm

[5] Jiang YC, Shi ZZ, Tang Y, Wang J. Fuzzy description logic for semantics representation of the semantic Web. Journal of Software, 2007,18(6):1257−1269 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/18/1257.htm

[6] Kang DZ, Xu BW, Lu JJ, Li YH. Reasoning within extended fuzzy description logic supporting terminological axiom restrictions. Journal of Software, 2007,18(7):1563−1572 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/18/1563.htm

[7] McIlraith S, Son T, Zeng H. Semantic Web services. IEEE Intelligent Systems, 2001,16(2):46−53.

[8] Shi ZZ, Dong MK, Jiang YC, Zhang HJ. A logical foundation for the semantic Web. Science in China (Series F: Information Sciences), 2005,48(2):161−178.

[9] Chang L, Shi ZZ, Qiu LR, Lin F. A tableau decision algorithm for dynamic description logic. Chinese Journal of Computers, 2008, 31(6):896−909 (in Chinese with English abstract).

[10] Chang L, Lin F, Shi ZZ. A dynamic description logic for representation and reasoning about actions. In: Zhang ZL, Siekmann J, eds. Proc. of the 2nd Int'l Conf. on Knowledge Science, Engineering and Management. Berlin: Springer-Verlag, 2007. 115−127.

© 中国科学院软件研究所 http://www.c-s-a.org.cn

[11] Shi ZZ, Chang L. Reasoning about semantic Web services with an approach based on dynamic description logics. Chinese Journal of Computers, 2008,31(9):1599−1611 (in Chinese with English abstract).

[12] Pratt VR. A practical decision method for propositional dynamic logic: Preliminary report. In: Proc. of the 10th Annual ACM Symp. on Theory of Computing. New York: ACM Press, 1978. 326−337. http://www.informatik.uni-trier.de/~ley/db/conf/stoc/stoc78.html #Pratt78

[13] Chou SC, Winslett M. A model-based belief revision system. Journal of Automated Reasoning, 1994,12(2):157−208.

[14] Bylander T. The computational complexity of propositional STRIPS planning. Artificial Intelligence, 1994,69(1-2):165−204.

[15] Giacomo GD, Lenzerini M. PDL-Based framework for reasoning about actions. In: Gori M, Soda G, eds. Proc. of the 4th Congresss of the Italian Association for Artificial Intelligence. Berlin: Springer-Verlag, 1995. 103−114.

[16] Reiter R. Knowledge in Action: Logical Foundations for Specifying and Implementing Dynamical Systems. Cambridge: MIT Press, 2001.

[17] Thielscher M. From situation calculus to fluent calculus: State update axioms as a solution to the inferential frame problem. Artificial intelligence, 1999,111(1-2):277−299.

[18] Baader F, Lutz C, Milicic M, Sattler U, Wolter F. Integrating description logics and action formalisms: First results. In: Veloso M, Kambhampati S, eds. Proc. of the 12th National Conf. on Artificial Intelligence. Menlo Park: AAAI Press /The MIT Press, 2005. 572−577.

[19] Gu YL, Soutchanski M. Decidable reasoning in a modified situation calculus. In: Veloso M, ed. Proc. of the 20th Int'l Joint Conf. on Artificial Intelligence. Menlo Park: AAAI Press, 2007. 1891−1897.

附中文参考文献:

[4] 李言辉,徐宝文,陆建江,康达周.支持数量约束的扩展模糊描述逻辑复杂性研究.软件学报,2006,17(5):968−975. http://www.jos.org.cn/1000-9825/17/968.htm

[5] 蒋运承,史忠植,汤庸,王驹.面向语义 Web 语义表示的模糊描述逻辑.软件学报,2007,18(6):1257−1269. http://www.jos.org.cn/1000-9825/18/1257.htm

[6] 康达周,徐宝文,陆建江,李言辉.支持术语公理约束的扩展模糊描述逻辑推理.软件学报,2007,18(7):1563−1572. http://www.jos.org.cn/1000-9825/18/1563.htm

[9] 常亮,史忠植,邱莉榕,林芬.动态描述逻辑的 Tableau 判定算法.计算机学报,2008,31(6):896−909.

[11] 史忠植,常亮.基于动态描述逻辑的语义 Web 服务推理.计算机学报,2008,31(9):1599−1611.

常亮(1980－),男,贵州赫章人,博士,CCF 学生会员,主要研究领域为描述逻辑,语义 Web,语义 Web 服务.

史忠植(1941－),男,研究员,博士生导师,CCF 高级会员,主要研究领域为人工智能,智能主体,机器学习.

陈立民(1982－),男,博士生,CCF 学生会员,主要研究领域为描述逻辑,语义 Web,形式化方法.

牛温佳(1982－),男,博士生,主要研究领域为智能主体,语义 Web 服务.